NASA-CR-202405

93/06

FINAL REPORT: MS Thesis

NASA GRANT NUMBER NCC 2-855
Cal Poly Project 5339

Subsonic Wing Optimization for Handling Qualities using ACSYNT

Dani Soban: Graduate Student

PI: Daniel J. Biezad

NASA Technical Monitor:
Mr. Paul Gelhausen
NASA Ames Research Center
Moffett Field, CA. 94035-1000

October 1996

PREDAVOR: DEVELOPMENT AND IMPLEMENTATION OF SOFTWARE FOR RAPIDLY ESTIMATING AIRCRAFT STABILITY DERIVATIVES AND HANDLING QUALITIES

A Thesis

presented to

the Faculty of the Aeronautical Engineering Department

California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Danielle Suzanne Soban

September 1996

APPROVAL PAGE

TITLE: PREDAVOR: DEVELOPMENT AND IMPLEMENTATION OF SOFTWARE FOR RAPIDLY ESTIMATING AIRCRAFT STABILITY DERIVATIVES AND HANDLING QUALITIES

AUTHOR: Danielle Suzanne Soban

DATE SUBMITTED: September 23, 1996

Dr. Daniel J. Biezad

Adviser ______________________ Signature

Dr. Russell Cummings

Committee Member ______________________ Signature

Dr. H. JoAnne Freeman

Committee Member ______________________ Signature

Abstract

The capability to accurately and rapidly predict aircraft stability derivatives using one comprehensive analysis tool has been created. The PREDAVOR tool has the following capabilities: rapid estimation of stability derivatives using a vortex lattice method, calculation of a longitudinal handling qualities metric, and inherent methodology to optimize a given aircraft configuration for longitudinal handling qualities, including an intuitive graphical interface. The PREDAVOR tool may be applied to both subsonic and supersonic designs, as well as conventional and unconventional, symmetric and asymmetric configurations. The workstation-based tool uses as its model a three-dimensional model of the configuration generated using a computer aided design (CAD) package. The PREDAVOR tool was applied to a Lear Jet Model 23 and the North American XB-70 Valkyrie.

Acknowledgments

I owe many thanks to those that provided insightful suggestions, answered numerous questions, and attended countless progress briefings as the product evolved. In particular **Paul Gelhausen**, who instigated the project; my adviser **Dr. Daniel J. Biezad** for his unfailing advice, encouragement and support; **J.R. Gloudemans** and **Dr. David Kinney**, for their help and support especially with the VORVIEW code; and the rest of the **Systems Analysis Branch** at NASA Ames Research Center for their support.

PREDAVOR is available upon request from the National Aeronautics and Space Administration. For more information, contact:

Dr. Daniel J. Biezad (805) 756-5126 dbiezad@oboe.calpoly.edu

Danielle Soban dsoban@daniel.aero.calpoly.edu

Table of Contents

Tables

Table | Page

List of Figures

CHAPTER 1

Introduction and Problem Summary

Justification

Traditionally, aircraft have been designed and built using a "design by discipline" approach. Each discipline, such as propulsion, structures, or aerodynamics, was optimized independently with minimal input from the other disciplines. Only after the aircraft design was fully determined were such disciplines ashandling analysis and economics considered. (Figure 1.1). Recently, however, advances in both technology and sophisticated analysis tools have spawned growing interrelationships and interdependencies within the various aerospace disciplines. For example, the use of composites links the disciplines of structures, aerodynamics, and controls together, and the effect of each of these upon the other must be considered during preliminary design and analysis. These new interdependencies and interrelationships have led to a new era in the aerospace industry, that of concurrent engineering (CE)[1]. Aerospace companies are moving towards an approach such as that shown in Figure 1.2, in which there is considerable interplay between the disciplines much earlier in the design process.

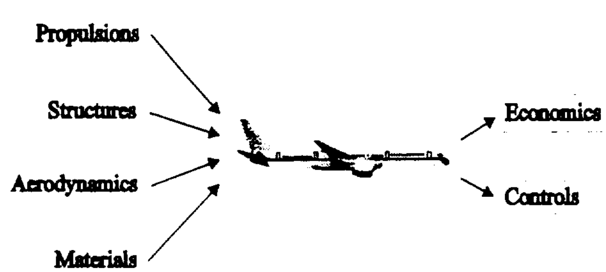


Figure 1.1

"Design by Discipline" Aircraft Design Approach

The advent of CE has led to the need for intuitive graphical design tools capable of multidisciplinary analysis. One such tool is ACSYNT (AirCraft SYNThesis), a workstation based modular optimization tool, the product of a government and industry institute that is administered by Virginia Polytechnic University[2]. The necessity of preliminary design tools such as ACSYNT is evident when considering the following. Although a relatively small fraction of life cycle costs are spent during the preliminary design phase of aircraft, mistakes and misjudgments during this phase can prove costly, and sometimes financially disastrous, to fix at later dates. If potential problems could be identified earlier in the design process, substantial time and money could be saved. Tools are therefore needed that model not only all of the disciplines themselves, but predict and establish the interrelationships of these disciplines. One such discipline not traditionally considered during the preliminary design phase of aircraft is the handling qualities and flight characteristics of the aircraft.

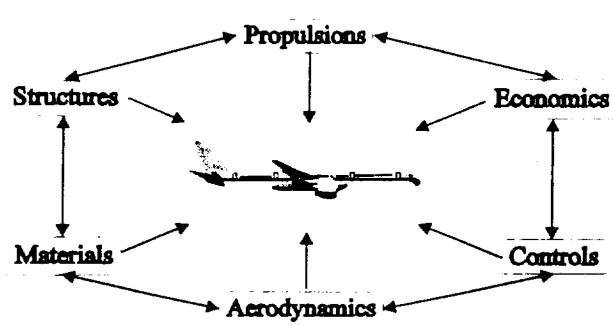


Figure 1.2

Concurrent Engineering Aircraft Design Approach

Studying the effects of handling qualities during the preliminary design phase has three primary advantages: reduction in cost, time, and complexity. The first consideration is cost. If an airplane has been designed to optimize its handling qualities, its inherent dynamics will minimize the risk and sophistication (complexity) of its control system, thus minimizing its cost. Concurrently, sensitivity studies conducted at the preliminary design phase of the aircraft could be used later in the development and testing process to study and understand any changes needed to the control system of the aircraft. This saves considerable time in the redesign phase of the aircraft, which is traditionally a very costly part of the program. Finally, if an analysis tool exists to examine the handling qualities of an aircraft at the preliminary design stage, data from this tool could be used in conjunction with other tools, such as a flight simulator, as a learning tool. In this way, both students of aeronautical engineering and industry engineers can get a rapid assessment of both the handling qualities of the aircraft itself, as well as how changes to the handling qualities affect other aspects of the design.

Statement of Problem

A tool, then, needs to be developed that is capable of predicting, analyzing, and optimizing the handling qualities and flight characteristics of an aircraft, including good estimations of its stability derivatives. Traditionally, empirical methods such as those found in USAF DATCOM[3] are used to predict these stability derivatives. Due to the empirical nature of these methods, reasonable accuracy is achieved for conventional designs. Yet the method considerably degrades when applied to asymmetrical or non-conventional designs. Since many of today's modern aircraft explore the concepts of unconventional and asymmetric flight, a method of analyzing them is a necessity. Recent advances in computing power have made the use of certain computational methods feasible. Vortex lattice methods are capable of generating data that may be used to calculate these stability derivatives. This method, in addition to being able to analyze asymmetrical and non-conventional designs, is also capable of providing data to calculate some derivatives that methods such as DATCOM are incapable of generating even for conventional designs. These include the control derivatives of the aircraft. Table 1.1 compares the capabilities of different methods to predict certain derivatives.

Table 1.1

A Comparison of the Capabilities of Other Methods to Predict Specific Aircraft Stability Derivatives

Derivatives	VORLAX/PREDAVOR	ACSYNT	DATCOM
C_L	X	X	X
$C_{L\alpha}$	X	X	X
$C_{L\dot{\alpha}}$	X	X	X
C_{Lq}	X		X
C_{Lu}	X	X	
C_D	X	X	X
$C_{D\alpha}$	X	X	
$C_{D\dot{\alpha}}$	X	X	
C_{Dq}	X		
C_{Du}	X	X	
C_M	X	X	
$C_{M\alpha}$	X	X	X
$C_{M\dot{\alpha}}$	X	X	X
C_{Mq}	X		X
C_{Mu}	X	X	
C_{MT}		X	
$C_{MT\alpha}$		X	
C_T		X	
C_{Txu}		X	
$Cl\beta$	X		X
$Cl\,p$	X		X
$Cl\,r$	X		X
$Cn\beta$	X		X
$Cn\,p$	X		X
$Cn\,r$	X		X
$Cy\beta$	X		X
$Cy\,p$	X		X
$Cy\,r$	X		
All Control	X		

When considering the handling qualities of an aircraft, a suitable metric of analysis needs to be selected. Several metrics were considered, such as classical Neal-Smith criteria, modern Neal-Smith, the bandwidth criteria, and the Control Anticipation Parameter (CAP)[4]. The CAP parameter was chosen for its ease of use, its intuitive nature, and its ability to be readily incorporated into the analysis code. It was also used to validate the optimization scheme. The framework established with the CAP parameter makes the future incorporation of more sophisticated metrics feasible.

Finally, a good analysis tool must be fast, easy to use, and readily understood. In addition, it must provide insight to users about the effects of their design decisions upon the flight characteristics of their aircraft. A workstation-based tool offers many advantages. First, a workstation can provide the computational power necessary for sufficient analysis. In addition, the operating environment of a workstation allows user-friendly and informative graphical interfaces (GUI's) to be created.

The PREDAVOR analysis tool was thus created and links the rapid estimation of stability derivatives with the automatic calculation of the CAP parameter. It does this by combining existing analysis tools with new code in order to create a consistent methodology for the analysis of aircraft and their flight characteristics. The PREDAVOR methodology was tested and comparisons were made between the derivatives generated by the method and empirically generated data, as well as some flight test data. The Lear Jet Model 23 aircraft was analyzed for optimization with respect to wing aspect ratio and horizontal tail longitudinal distance. In order to validate the method for supersonic flight conditions, stability derivatives for the North American XB-70 were generated for both subsonic and supersonic conditions.

CHAPTER 2

Existing Software Used in Development of PREDAVOR

The PREDAVOR methodology uses as its foundation the capabilities of several existing tools. The input and output of these tools are then linked together with new code to produce an overall methodology.

The advantages and disadvantages of using existing code, rather than developing completely new code, were examined carefully when planning the PREDAVOR framework. Using existing tools eliminated the need to duplicate effort. It makes little sense to write code to perform a task when such a code already exists. In addition, it can be assumed that an existing code is further along in its validation process, and thus more robust. The chief disadvantage to using several different codes is linking the codes together in a cohesive manner. Different codes imply different input and output format, different programming languages, and potentially different operating environments.

In this particular case, two primary codes were heavily in use prior to the project development. The decision was made to use these codes as the foundation for PREDAVOR, and to link the software packages together using new code.

ACSYNT

The ACSYNT aircraft design code is used to generate the wireframe model used in the PREDAVOR analysis. The workstation-based ACSYNT (AirCraft SYNThesis) is modular in design. Each discipline, such as aerodynamics, weights, or economics, is contained in an individual module, and these modules are linked together through an analysis package. The code is capable of analyzing a wide variety of aircraft including civil and military aircraft, fighters, bombers, and transports. The modular components of ACSYNT allow analysis of a single discipline, or the modules can be combined in order to evaluate the integrated results[1]. Currently, ACSYNT is administered by Virginia Polytechnic University[2]. Originally, however, ACSYNT was developed by NASA Ames Research Center for conceptual design studies of advanced aircraft and is still heavily in use today.

The real power of ACSYNT lies in its non-linear optimization code. This methodology allows the vehicle to be optimized for a particular objective function or functions (such as gross takeoff weight), given various restraints. In order for the non-linear optimization code to be as realistic and feasible as possible, it is important for all of the components of the synthesis process to be modeled correctly. For this reason, the modules in ACSYNT are parameter driven with equations derived from theory as opposed to table look-up methods[3]. It is future goal of this project to use this optimization package to automate the handling qualities optimization scheme.

The version of ACSYNT currently being used in the PREDAVOR project includes a CAD interface written entirely in the three-dimensional graphics standard PHIGS (Programmer's Hierarchical Interactive Graphics System)[4]. This CAD package allows a model of the aircraft to be rapidly constructed using component templates (see Figure 2.1.) Once the model is completed, it may be easily transferred into a file format that can be used by the other codes in the PREDAVOR methodology.

VORLAX

PREDAVOR uses a vortex lattice method called VORLAX to generate the forces and moments on the model that are used to calculate the stability derivatives. Variations of the basic vortex lattice method are currently being used to analyze both planar and non-planar aircraft configurations. The beauty of the vortex lattice method lies in the simplicity of its numerical technique as well as its high degree of accuracy (within the limits of the basic theory)[5].

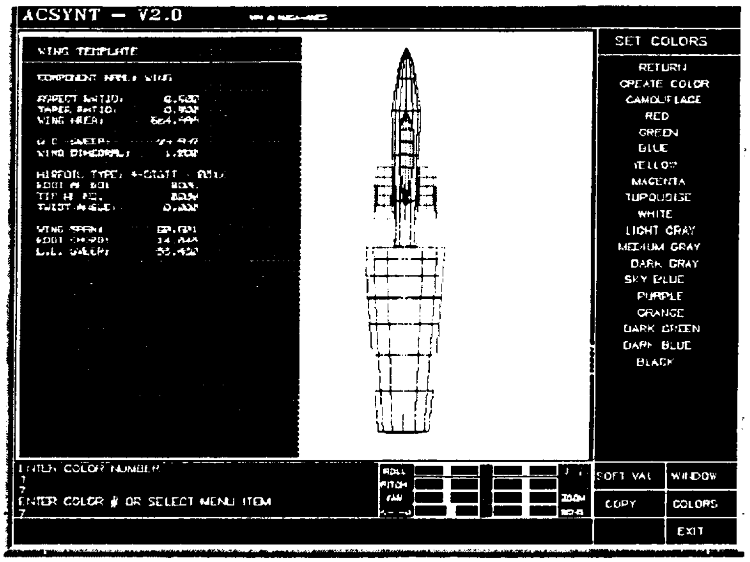
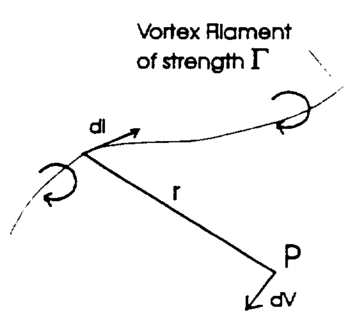


Figure 2.1.

ACSYNT Screen and Model with Wing Template

Basic Vortex Lattice Theory

The basic vortex lattice method involves superimposing a finite number of horseshoe vortices of different strengths Γ_n on to the surface of the model. Consider, for example, part of a finite wing shown in Figure 2.2. A horseshoe vortex (abcd) of strength Γ_n is placed upon a representative trapezoidal panel. The velocity induced at an arbitrary point P(x,y) by this single horseshoe vortex can be calculated using the Biot-Savart Law:



$$dV = \frac{\Gamma}{4\pi} \frac{\mathbf{dl} \times \mathbf{r}}{|\mathbf{r}|^3}$$

In order to analyze an aircraft (or any other shape), the entire surface is replaced with a series of representative trapezoids (Figure 2.3). A horseshoe vortex is then placed on each trapezoid. The total induced velocity at point P(x,y) may again be found using Biot-Savart. By applying the flow tangency condition to all control points, a system of simultaneous equation may be obtained and solved for the unknown circulations (Γ_n's)[6]. These, in turn, directly correspond to the forces and moments acting upon the model.

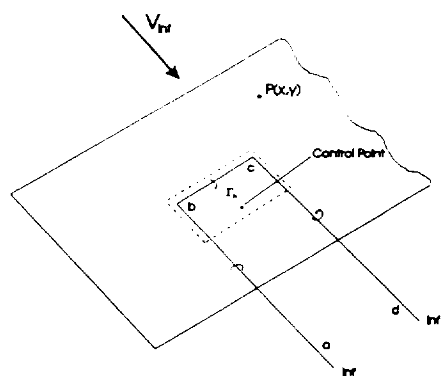


Figure 2.2

Schematic of a Single Horseshoe Vortex

Source: Anderson, J. D. Fundamentals of Aerodynamics. New York: McGraw-Hill, 1991.

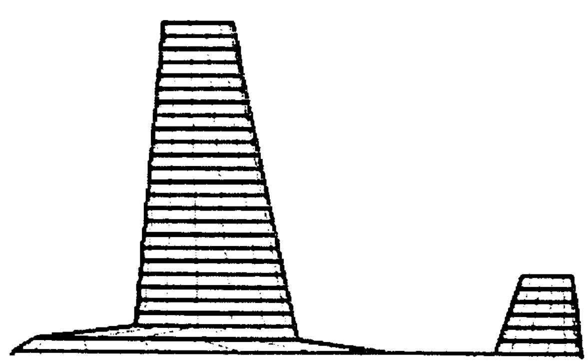


Figure 2.3

Trapezoidal Half-Model of Aircraft

VORLAX Code

Although vortex lattice methods are currently being used and have proved to be practical and versatile tools, most analysis has been largely subsonic. The applicability of the basic techniques of vortex lattice theory to supersonic flow has been largely ignored[7]. VORLAX, developed by Lockheed in 1977, is applicable to both subsonic and supersonic flight conditions. The supersonic capability is justified as follows. Assume that the discrete vortex lattice approximates the vorticity on the surface of the model. The mathematical representation of this includes an integral that has a residual term of the velocity field. Using this residual term correctly by including it in the resulting velocity field generated by the vortex lines, allows the calculation, and thus applicability, for supersonic flow[8].

In addition, the VORLAX method includes special techniques for simulating the thickness of lifting surfaces using a double (bi-planar) vortex lattice layer. VORLAX is also capable of analyzing fusiform bodies by arranging a vortex grid on a series of concentric cylindrical surfaces. These concepts are all illustrated in Figure 2.4 which shows a generalized vortex lattice model of a wing-body configuration.

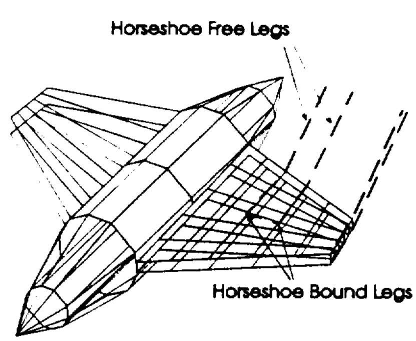


Figure 2.4

Generalized Vortex Lattice Model of Wing-Body Configuration

Source: Recreated from Miranda, L. R., and R.D. Elliot and W. M. Baker. "NASA CR-2865 A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications," NAS11-12972. Dec. 1977.

VORVIEW

In order to facilitate the rather complex input to VORLAX, NASA Ames has developed a graphical pre-processor to the code, called VORVIEW[9]. The input to VORLAX had consisted of lengthy files that numerically defined the coordinates of each trapezoid, as well as other required information for analysis. There was no visual feedback of the model being analyzed, and changes to the model were manual and tedious.

VORVIEW, on the other hand, uses as its input the wireframe geometry generated by ACSYNT (Figure 5). This file, together with a data file containing flight conditions, is used to launch VORVIEW. The wireframe model may then be "sliced" from wing tip to wing tip, and subdivided into trapezoids. Instead of defining each trapezoid numerically, as the input to VORLAX requires, VORVIEW allows the trapezoids to be created graphically and the manual input file to VORLAX created automatically.

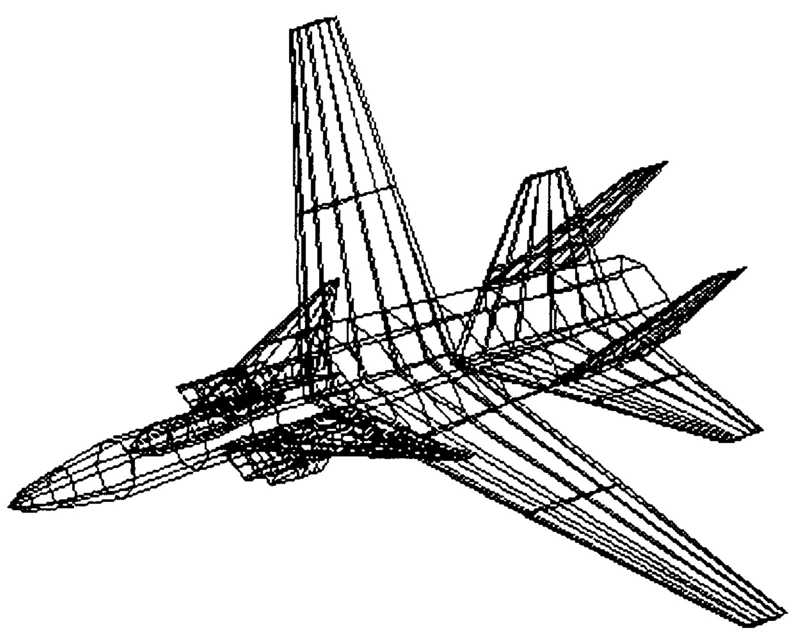


Figure 2.5

Wireframe Model Generated by ACSYNT and used by VORVIEW

While VORVIEW does a nice job slicing the model in the planform view, it does not currently posses the capability to create vertical surfaces automatically. These panels can, however, be created by hand. Figure 2.6 shows a sliced and subdivided VORVIEW model.

After the model has been sliced and subdivided, VORVIEW transforms the data and runs VORLAX. The output is shown both graphically as a Cp distribution (Figure 2.7) and numerically as forces and moments in an output file.

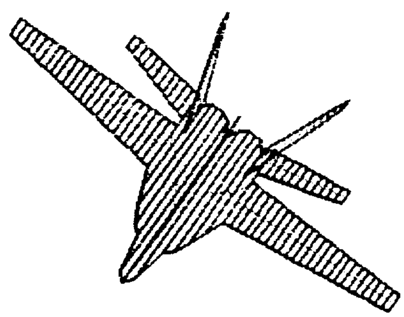


Figure 2.6

Sliced and Subdivided VORVIEW Model

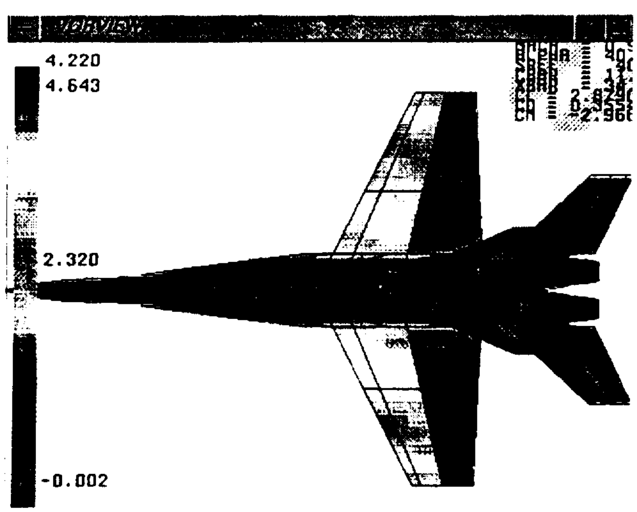


Figure 2.7

VORVIEW Output Showing Cp Distribution

CHAPTER 3

The PREDAVOR Code

Methodology Before PREDAVOR

Although rather tedious, the tools of the previous section could be used in succession to generate the stability derivatives of a given model. The methodology would be as follows:

1. Create a model using ACSYNT.
2. Edit the input file for initial flight conditions.
3. Run VORVIEW/VORLAX.
4. Manually parse out the resulting forces and moments from the output file.
5. Edit the input file to contain a perturbation of the flight conditions. (For example, change $\alpha = 0$ deg to $\alpha = 2$ deg.)
6. Re-run VORVIEW/VORLAX.
7. Manually parse out the new results.
8. Manually calculate the stability derivative from the results of both runs.
9. Edit the input file to undo the perturbation.
10. Repeat steps 1-9 for each derivative.

In order to generate a complete set of derivatives, the VORVIEW/VORLAX combination would need to be run once at unperturbed conditions, and once for each perturbation needed (alpha, beta, pitch rate, yaw rate, roll rate, control surface deflections, and change in forward velocity. The results of each of these runs must be parsed, and each derivative calculated by hand. Thus, to generate a standard set of derivatives, many runs of VORVIEW/VORLAX must be made and many sets of manual calculations performed. The entire process must be repeated if analysis is needed at a different flight condition.

While it is certainly possible to generate sets of stability derivatives in the above manner, it is not practical. PREDAVOR was designed to automate this process. This has several advantages. The first is the elimination of tedious hand calculations involving multiple runs of the code, manual manipulation of the input files, parsing large output files, manual axes transformations, and the calculations of the stability derivatives themselves. Secondly, accuracy may be improved through the elimination of many sources of human error. Thirdly, time is saved through multiple autonomous runs of the VORLAX code. And finally, by automating this process, it is possible to one day incorporate the PREDAVOR methodology into a mathematical optimization scheme, such as COPES/CONMIN associated with the ACSYNT package.

PREDAVOR Architecture

Fig 3.1 illustrates the overall PREDAVOR architecture. The first step is the creation of the three-dimensional wireframe model using the CAD package in ACSYNT. Next, generic flight conditions and a few basic geometric parameters are added to the VORVIEW input file. The graphical pre-processor VORVIEW is then used.

VORVIEW's current capabilities allow the user to slice the planform view of the aircraft from wing tip to wing tip. In order to calculate the lateral derivatives, however, a model of the vertical surfaces needs to be included. These vertical panels may be created manually by editing a supplementary file that includes the geometric slice data. The user simply adds the X, Y, and Z locations of each of the four points of the trapezoid to be created to the file. VORVIEW allows the newly created trapezoid to be viewed graphically. Figure 3.2 shows a three-view of a model created in this manner.

Once a satisfactory slice model is created, VORVIEW is run once to create the appropriate input file to VORLAX. Once this file is created, PREDAVOR edits it automatically, allowing multiple runs of VORLAX to be performed independently of VORVIEW.

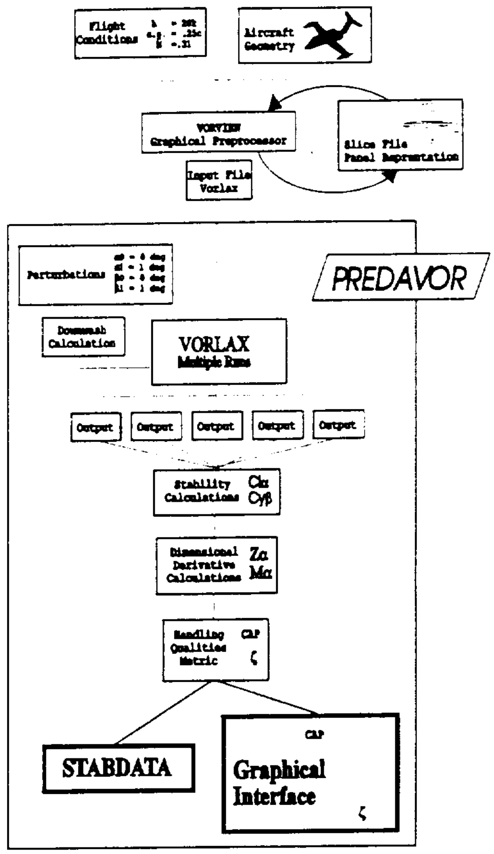


Figure 3.1

PREDAVOR Code Architecture

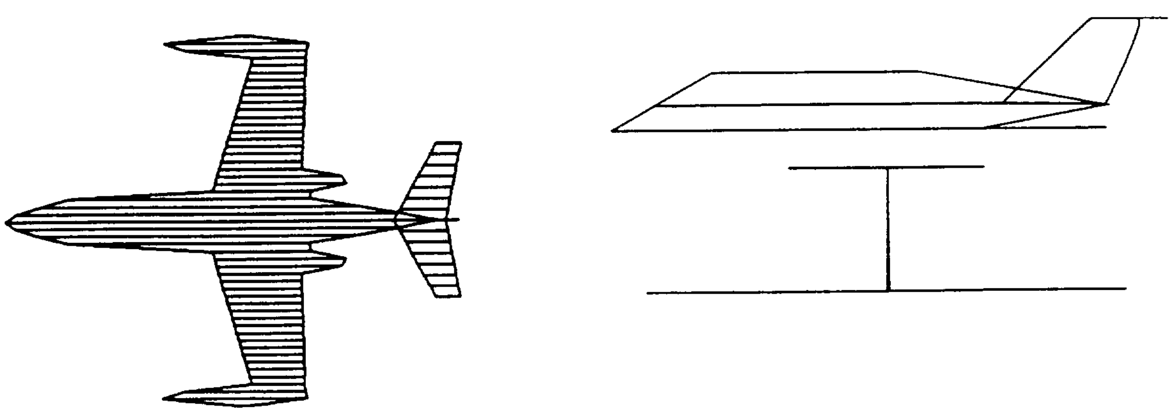


Figure 3.2

VORLAX Model with Manually Created Vertical Panels

At this point the user may add control surfaces to the aircraft. This is done via a control surface menu in VORVIEW. A planform of the sliced aircraft is shown, and control surfaces added by clicking on the appropriate panel. Control surface type, per cent chord length, and deflection angle are all inputs. The control surface part of VORVIEW was modified to allow separate input files to be created for each control surface (Figure 3.3). A toggle button allows the user to choose between elevator, aileron, and "other". The control surface is created using a point and click technique, and the user presses the "SET INPUT" button to create the new control surface input file. Because the control surface process in VORVIEW works only from a planform view, rudders may not be created explicitly in this manner. The "other" option was created to anticipate VORVIEW's future ability to create vertical panels automatically. Until then, the user simply creates a deflected rudder manually, using the method described earlier to create vertical panels by hand. The derivative may be calculated using the steps outlined at the beginning of this chapter.

The next step for the user is to edit the PREDAVOR input file to include the proper flight conditions, baseline flight variable values, and perturbed conditions. This flight conditions file, together with the input file(s) created by VORVIEW, are used to run the PREDAVOR code.

PREDAVOR makes multiple runs through VORLAX, changing its input file automatically to reflect the necessary perturbations. PREDAVOR sifts through the rather large output data files and parses out the necessary data. The stability derivatives are calculated, along with the dimensional derivatives, and the handling qualities parameter CAP. Options exist to calculate the downwash due to the horizontal tail, and to perform the transformation from wind axes to body axes.

PREDAVOR may be used in a manual handling qualities optimization scheme. Geometric changes to the model may be made, and the process to

At this point the user may add control surfaces to the aircraft. This is done via a control surface menu in VORVIEW. A planform of the sliced aircraft is shown, and control surfaces added by clicking on the appropriate panel. Control surface type, per cent chord length, and deflection angle are all inputs. The control surface part of VORVIEW was modified to allow separate input files to be created for each control surface (Figure 3.3). A toggle button allows the user to choose between elevator, aileron, and "other". The control surface is created using a point and click technique, and the user presses the "SET INPUT" button to create the new control surface input file. Because the control surface process in VORVIEW works only from a planform view, rudders may not be created explicitly in this manner. The "other" option was created to anticipate VORVIEW's future ability to create vertical panels automatically. Until then, the user simply creates a deflected rudder manually, using the method described earlier to create vertical panels by hand. The derivative may be calculated using the steps outlined at the beginning of this chapter.

The next step for the user is to edit the PREDAVOR input file to include the proper flight conditions, baseline flight variable values, and perturbed conditions. This flight conditions file, together with the input file(s) created by VORVIEW, are used to run the PREDAVOR code.

PREDAVOR makes multiple runs through VORLAX, changing its input file automatically to reflect the necessary perturbations. PREDAVOR sifts through the rather large output data files and parses out the necessary data. The stability derivatives are calculated, along with the dimensional derivatives, and the handling qualities parameter CAP. Options exist to calculate the downwash due to the horizontal tail, and to perform the transformation from wind axes to body axes.

PREDAVOR may be used in a manual handling qualities optimization scheme. Geometric changes to the model may be made, and the process to this point repeated. A flag in the PREDAVOR input files allows all handling qualities data to be concatenated to a single file, until the flag is changed. The CAP graphical interface then uses this information to create a CAP plot and presents it to the user, allowing them to identify handling qualities trends and optimize their aircraft to the results.

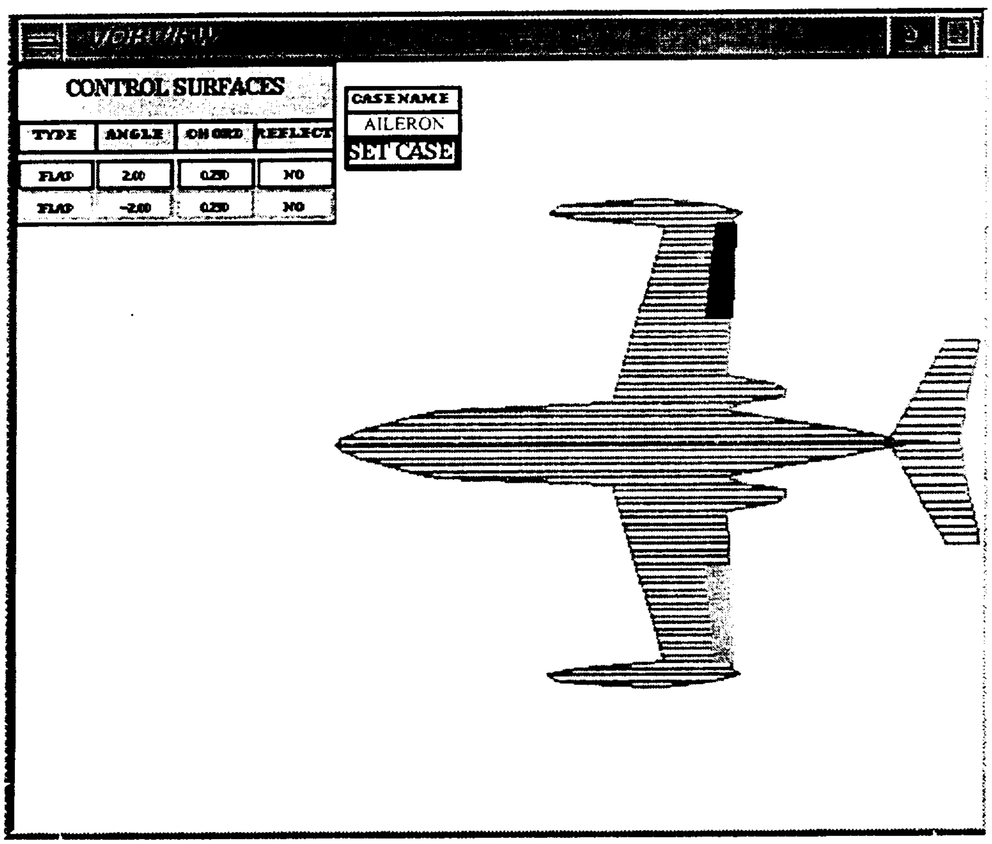


Figure 3.3

VORVIEW Screen Shot Showing Addition of Control Surfaces

Input Files

The goal of project PREDAVOR is to rapidly estimate stability derivatives using given existing tools. Automating as much of the

Operating Environment

Both ACSYNT and VORVIEW were designed to operate on Silicon Graphics (SGI) workstations, optimally running IRIX version 4.0.2. ACSYNT is written mostly in FORTRAN, while VORVIEW is primarily written in ANSI C. Both, however, have graphical interfaces that are compatible with the SGI's. PREDAVOR, in order to ensure compatibility, was written in ANSI C and runs on the SGI workstations.

It must be noted, however, that the only part of the PREDAVOR process that requires graphical, workstation abilities is the creation of the model and the initial run of VORVIEW. Once these steps are completed, a user may download the necessary files to any system that is capable of running compiled C code. The rest of the process and the analysis may then be completed on the new system.

PREDAVOR Calculations

In addition to editing input files, performing multiple VORLAX runs, and parsing output data, PREDAVOR performs internal calculations to generate the stability derivatives, the dimensional derivatives, and axes transformations.

Stability Derivative Calculation

The output of VORLAX contains the total forces and moments upon the analyzed model. These forces and moments are in turn used to calculate the non-dimensional stability derivatives of the model at that flight condition. Usually, stability derivative data, such as flight test data, wind tunnel results, and theoretical computations, are given in non-dimensional stability derivatives. This facilitates comparison of aerodynamic characteristics of different aircraft as well as those of the same aircraft at different flight conditions[2]. The stability derivatives generated by PREDAVOR are thus of the non-dimensional form.

An example of a stability derivative calculation is as follows. Each derivative is non-dimensionalized as appropriate.

CHAPTER 4

The PANGLOSS Project

The PREDAVOR methodology is part of a larger framework called the PANGLOSS Project. The goal of PANGLOSS is to provide students with accurate yet intuitive tools that would allow them to rapidly analyze and understand aircraft stability, control, and handling qualities. PANGLOSS is an ongoing project at Cal Poly and team members consist mostly of graduate students designing analysis tools to be used at the undergraduate level.

One major branch of PANGLOSS is comprised of three projects that are designed to work together in a seamless methodology. PREDAVOR is an important part of this branch. The framework of this branch is shown in Figure 4.1. In the upper left hand corner a burgeoning aerospace design engineer conceives of an aircraft design. First they models their aircraft and obtain stability derivatives as well as a first cut handling qualities analysis from PREDAVOR. Next, they can analyze and manipulate this data using the intuitive graphical interface SAVI. Finally, they can input this new data, gained from PREDAVOR and SAVI, into a workstation-based simulator called RADIAN. In this way, the designer can very rapidly conceive of a design, analyze it, and actually fly his design, all in a matter of hours.

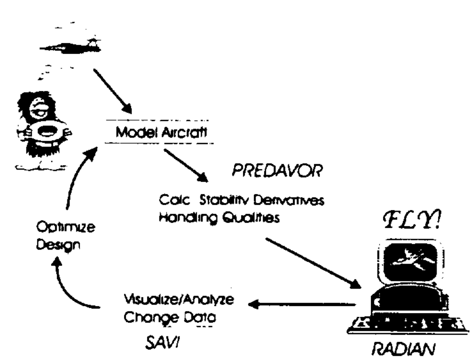


Figure 4.1

PANGLOSS Project Overview

Project PREDAVOR is dealt with extensively in Chapter 3 of this document. The following sections briefly summarize and highlight Projects SAVI and RADIAN.

SAVI

Project SAVI[2] was conceived as a way to give designers direct access to simulator data in an easy to understand format. Most simulators use table look-up methods. These tables consist of thousands of data points. If the aircraft designer wishes to analyze or manipulate any of this data, he must stop the simulator, identify the data points he wishes to change, an edit the files using a standard text editor. The data must then be reloaded into the simulator and the simulator started. SAVI allows the designer access to the table information in intuitive graphical interfaces. Figure 4.2 shows the SAVI control window and Figures 4.3 and 4.4 show a 2-D and 3-D plot. The 3-D plot may be rotated for better viewing.

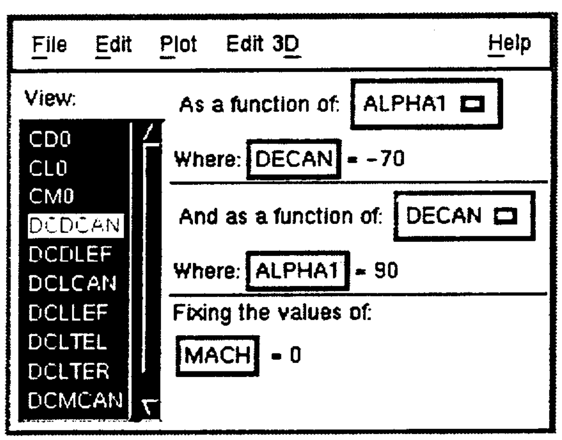


Figure 4.2

SAVI Control Window

Once viewed, the regions of data of interest may then be isolated using point and click methods. Data can be changed by clicking and dragging on a data point or by entering a new value. All data is accessed directly into simulator memory so these changes can be made while the simulator is active. The aircraft in the simulator immediately reacts with the new dynamics.

SAVI contains the following features:

- Editing algorithms for one or more points in 2-D and 3-D plots.
- Input from data file or direct interface with simulation memory
- Based on generic C and X-windows for portability
- Implemented with Motif libraries for consistent look and feel
- Postscript output for hard copy of plots
- HTML on-line user's manual
- Direct interface to simulation memory

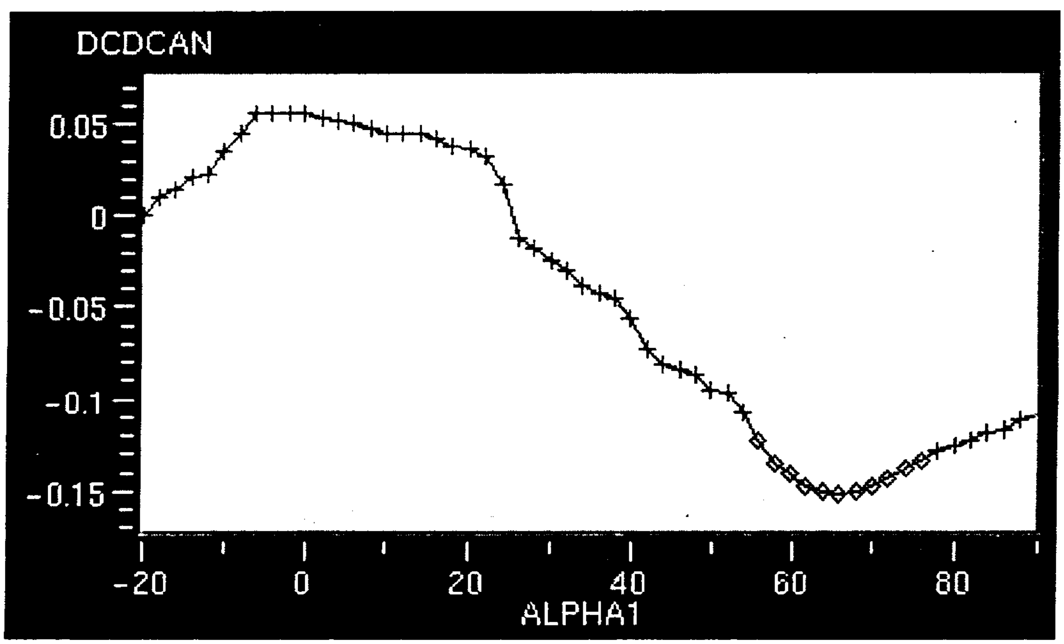


Figure 4.3

SAVI Two-Dimensional Plot Window

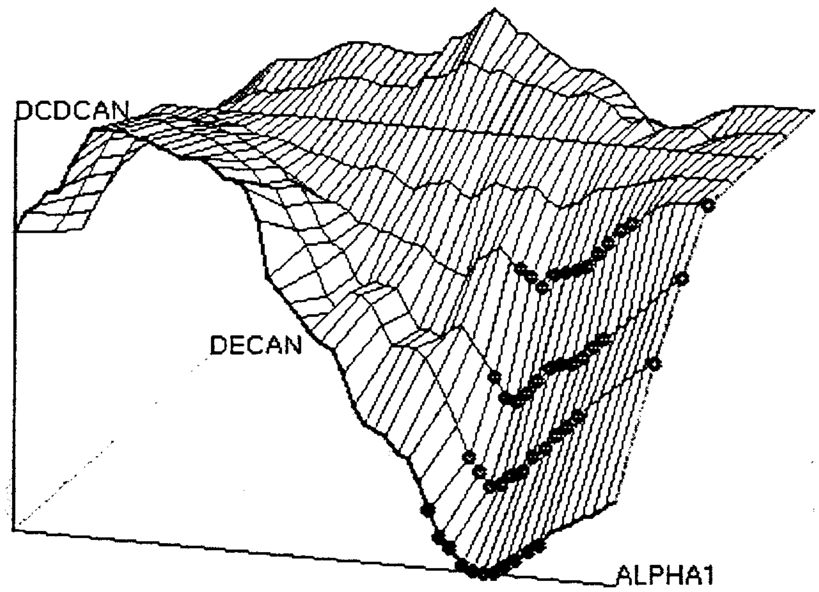


Figure 4.4

SAVI Three-Dimensional Plot Window

RADIAN

Project RADIAN[2] consists of the development of a workstation-based flight simulator that will have the following features:

- Six degrees of freedom simulator.
- Full non-linear equations of motion.
- Workstation-based, flight stick or mouse.
- Performance evaluation consisting of an "up and away" task and a landing task.
- Visual representation of model on screen.

The simulator will use data generated by PREDAVOR and SAVI.

RADIAN contains two performance evaluation situations that allows the designer to qualitatively evaluate the aircraft dynamics. The up and away task, shown in Figure 4.5, consists of a floating cross with a light on one end. The light changes locations on the cross in a random fashion. The goal of the pilot is to point the aircraft nose directly at the light. The pilot gains a score that corresponds to his success. The algorithm for this feature is still in progress. The second task is the landing task, shown in Figure 4.6. The pilot lands the aircraft and gains a score based on, among other variables, the rate of descent at touchdown. The pilot is aided by a vertical slope indicator in the form of "telephone poles". When the poles are level, the aircraft is on the flight path.

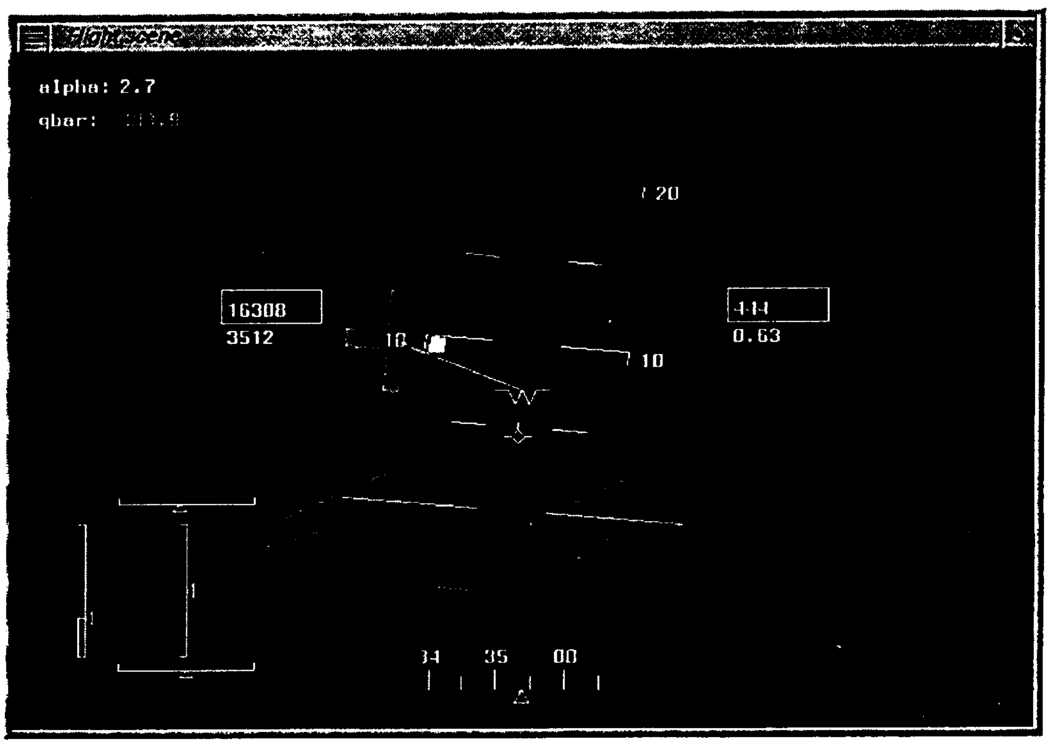


Figure 4.5

RADIAN Up and Away Simulator Task

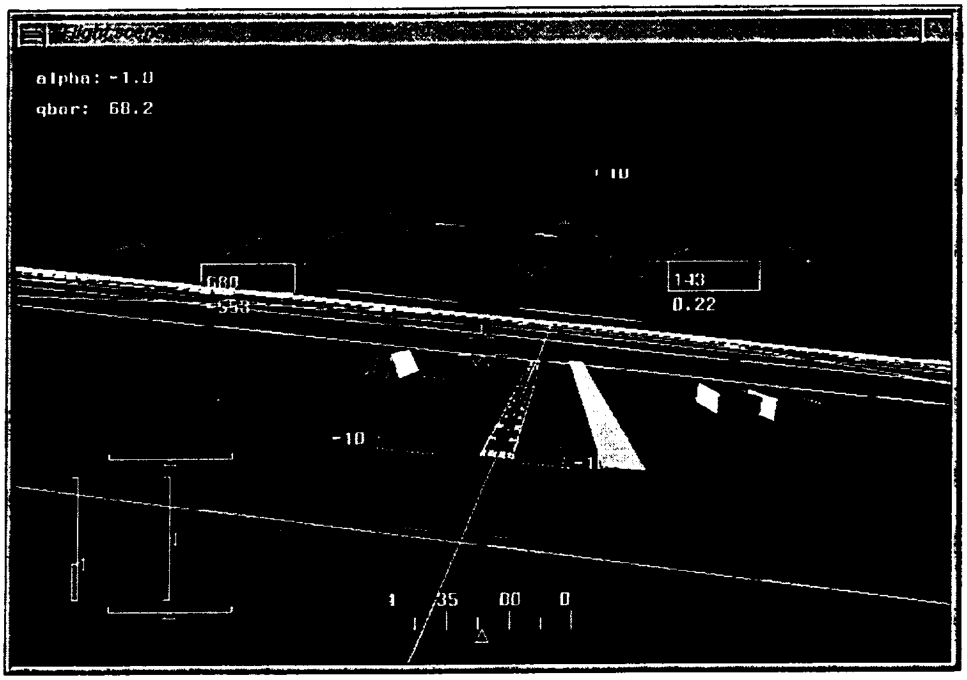


Figure 4.6

RADIAN Landing Simulator Task

Future Work

At the time of this writing, Projects PREDAVOR and SAVI are completed and working in a stand-alone fashion. The RADIAN simulator is still under construction. When finished, the three independent codes need to be integrated into a seamless methodology and tested thoroughly for robustness of method. Other PANGLOSS projects include Matlab-based packages for investigating handling qualities of aircraft, and a PC-based code to take aircraft geometry and determine state space matrices.

Project PREDAVOR is dealt with extensively in Chapter 3 of t s document. The following sections briefly summarize and highlight Projects SAVI and RADIAN.

SAVI

Project SAVI was conceived as a way to give designers dir t access to simulator data in an easy to understand format. Most simulators use table look-up methods. These tables consist of t us of data points. If the aircraft designer wishes to analyze or manipulate any of this data, he must stop the simulator, identi: t data points he wishes to change, an edit the files using a stan· .u editor. The data must then be reloaded into the simulator and e simulator started. SAVI allows the designer access to the tabl information in intuitive graphical interfaces. Figure 4.2 shov the SAVI control window and Figures 4.3 and 4.4 show a 2-D and 3-D .o The 3-D plot may be rotated for better viewing.

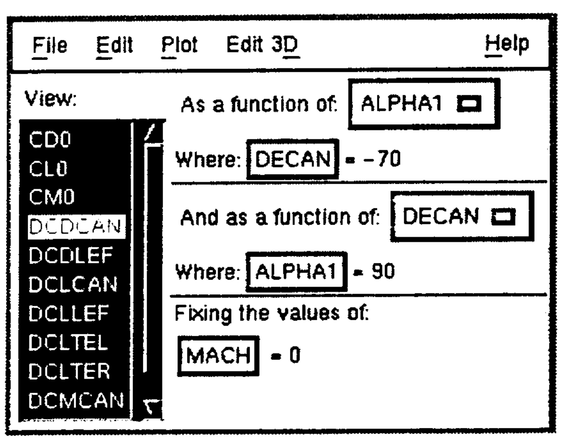


Figure 4.2

SAVI Control Window

Once viewed, the regions of data of interest may then b using point and click methods. Data can be changed by click ng dragging on a data point or by entering a new value. All d .

CHAPTER 5

Testing and Results

In order to validate the PREDAVOR methodology, test cases were conducted. For the subsonic case, a Lear Jet Model 23 was used, and for the supersonic case, the North American XB-70 Valkyrie was selected. Both models were chosen because stability derivative data as well as geometric data was readily available. In addition, a basic handling qualities analysis was conducted.

Subsonic Case- Lear Jet Model 23

PREDAVOR was applied to a conventional subsonic aircraft, the Lear Jet Model 23. This T-tail aircraft features fuselage mounted engines as well as fuel tip tanks. The aircraft model, shown in Figure 5.1, was created using ACSYNT. The aircraft was analyzed at the flight conditions shown in Table 5.1.

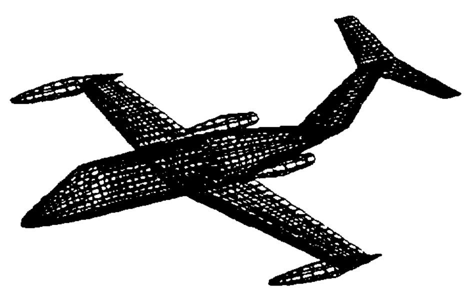


Figure 5.1

Wireframe Model of Lear Jet Model 23

The planform model of the aircraft was "sliced" automatically using the VORVIEW interface to create the analysis panels. Vertical panels were created by hand. The slice model is shown in Figure 5.2.

Table 5.1- Flight Conditions for Lear Jet Model 23

Flight Condition	**Cruise Max. Weight**
Altitude (ft)	40,000
Air Density (slugs/	.000588
Speed (fps)	677 (M=0.7)
Initial Attitude (deg)	2.7
Geometry and Inertias	
Wing Area (ft^2)	231.77
Wing Span (ft)	34.1
Wing Geo. Chord (ft)	7.03
Weight (lbs)	13,000
I_{xxB} (slug ft^2)	28,000
I_{yyB} (slug ft^2)	18,800
I_{zzB} (slug ft^2)	47,000
I_{xzB} (slug ft^2)	1,300

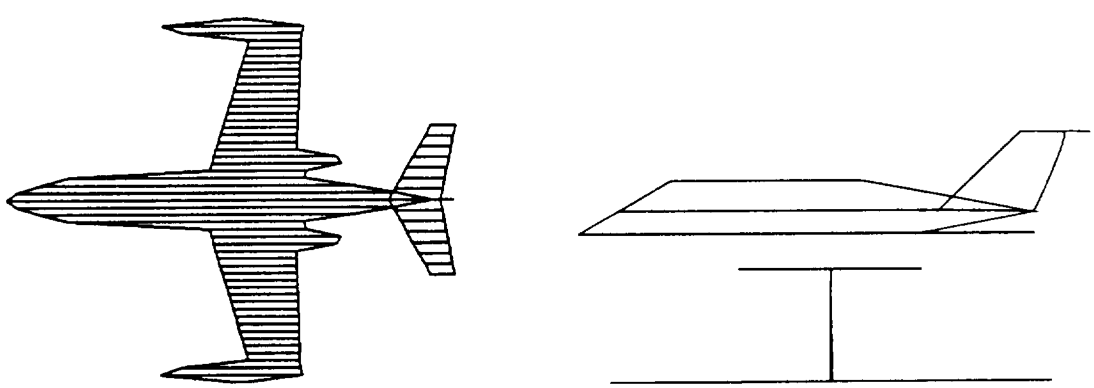


Figure 5.2- Sliced Representation of Lear Jet Model 23

The model of the Lear Jet was then analyzed using 150 wing tip to wing tip slices and 1500 subpolygons. The resulting stability derivatives are shown in Table 5.2. The derivatives were compared to those generated using empirical methods for the same aircraft at the given flight conditions[1]. Included in the table are relative importance of the derivatives[4]. The estimated accuracy using the empirical method is given in order to facilitate a comparison.

Table 5.2- Stability Derivatives of Lear Jet Model 2

Longitudinal Stability Derivatives

Derivatives	VORLAX	Emp. Data	Importance *	Est. Pred.
C_L	0.2594	0.4100		
$C_{L\ a}$	5.50	5.84	10	±5%
$C_{L\ a\ dot}$	2.98	2.20	4	±40%
$C_{L\ q}$	9.93	4.70	3	±20%
$C_{L\ u}$	8.37	0.40	5	±20%
C_D	0.0261	0.0335		
$C_{D\ a}$	-0.3723	0.3000	5	±10%
$C_{D\ u}$	-0.0644	0.1040	6	±20%
C_M	-0.0247	0.00		
$C_{M\ a}$	-0.5701	-0.6400	10	±10%
$C_{M\ a\ dot}$	-4.9660	-6.700	7	±40%
$C_{M\ q}$	-16.55	-15.50	9	±20%
$C_{M\ u}$	-1.7991	0.050	8	±20%

Lateral Stability Derivatives

Derivatives	VORLAX	Emp Data	Importance *	Est. Pred.
Cl b	-0.3849	-0.1100	10	±20%
Cl p	-0.4818	-0.4500	10	±15%
Cl r	0.2252	0.1600	7	±40%
Cn b	0.5999	0.1270	10	±15%
Cn p	-0.0797	-0.0080	8	±90%
Cn r	-0.5475	-0.2000	9	±25%
Cy b	-2.4666	-0.7300	7	±20%
Cy p	0.1759	0.0000	4	±50%
Cy r	1.3567	0.4000	4	±30%

*Relative Importance, 10=Major, 5=Minor, 0=Negligible, Roskam

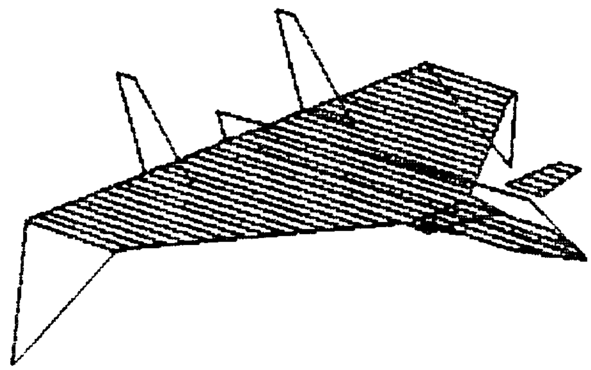


Figure 5.4

Slice Model of the XB-70

Table 5.3- Flight Conditions for the XB-70

Flight Condition	**Cruise Max. Weight**
Altitude (ft)	60,000
Air Density (slugs/	.0002237
Speed (fps)	2420 (M=2.5)
Initial Attitude (deg)	4.4
Geometry and Inertias	
Wing Area (ft^2)	6297.8
Wing Span (ft)	105
Wing Geo. Chord (ft)	78.53
Weight (lbs)	13,000
I_{xxb} (slug ft^2)	.18E7
I_{yyb} (slug ft^2)	.10E8
I_{zzb} (slug ft^2)	.221E8

The stability derivatives for the XB-70 were calculated and are tabulated in Table 5.4. The derivatives for the most part agree with data from various sources, including flight test data[3]. Of the important derivatives, Cl_β and Cn_β are again overpredicted, but still well within tolerable range. This agreement illustrates the vortex lattice code's ability to analyze supersonic configurations. In both the subsonic and supersonic case, it was found that this method is extremely sensitive to the placement of the center of gravity. Handling qualities analysis showed that the XB-70 is a Level 1 aircraft at both the subsonic and supersonic conditions tested. Optimization studies are in progress.

Table 5.4- Stability Derivatives of the XB-70

Longitudinal Stability Derivatives

Derivatives	VORLAX	Data**	Importance *
C_L	0.08	0.091	
$C_{L\ a}$	1.13	1.50	10
$C_{L\ a\ dot}$	-		4
$C_{L\ q}$	0.709		3
$C_{L\ u}$	-1.88		5
C_D	-		
$C_{D\ a}$	-		5
$C_{D\ u}$	0.0002		6
C_M	-		
$C_{M\ a}$	-0.155	-0.14	10
$C_{M\ a\ dot}$	0.017	0.0	7
$C_{M\ q}$	-0.565	-0.4	9
$C_{M\ u}$	0.469		8

Lateral Stability Derivatives

Derivatives	VORLAX	Data**	Importance *
Cl b	0.005	0.013	10
Cl p	-.065	-0.07	10
Cl r	-0.049	-0.015	7
Cn b	0.097	0.05	10
Cn p	-0.048	-0. 075	8
Cn r	-0.089	-0.36	9
Cy b	-0.23	-0.36	7
Cy p	0.11		4
Cy r	0.20		4

*Relative Importance, 10=Major, 5=Minor, 0=Negligible, Roskam

**Source: Heffley, R. K, and W.F. Jewell."NASA CR 2144 Aircraft Handling Qualities," December 1972.

Optimization of Wing and Horizontal Tail

The optimization scheme was applied to the geometry of the Lear Jet Model 23 by varying horizontal tail location and aspect ratio. Results are shown in Figure 5.5. First, the longitudinal location of the horizontal tail was changed. Point 5 on the graph locates the actual position of the horizontal tail. The tail was then moved forward and aft in 3 foot increments. At its original location, the Lear Jet is a Level 1 aircraft to Category B tasks. As the tail is moved fore, the aircraft moves away from the Level 1 space, with both CAP and ζ increasing. As the tail approaches the moment center of the aircraft, the handling qualities stay solidly Level 1.

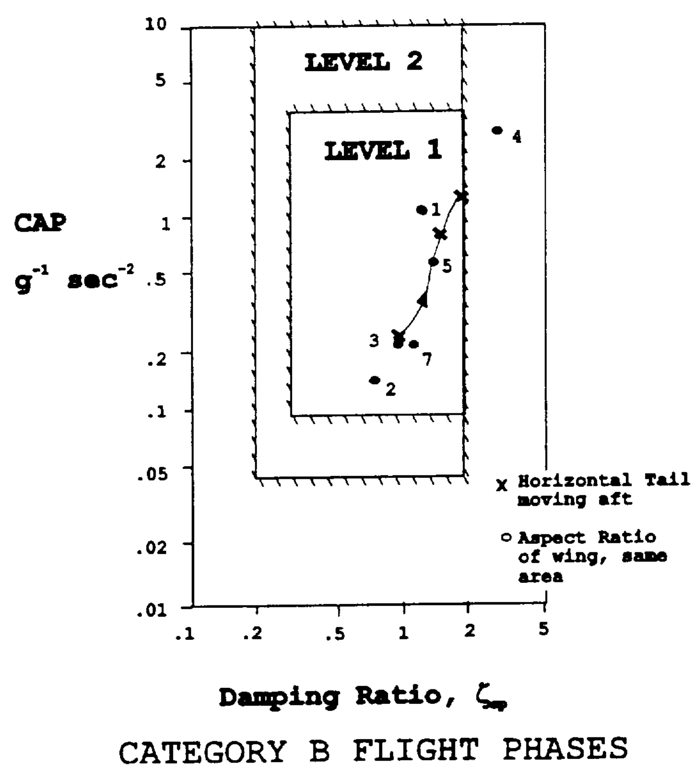


Figure 5.5

CAP Graph for Lear Jet Model 23

Next, the aspect ratio was varied, keeping constant wing area and allowing the wingspan to change. The points on the CAP graph are numbered with the value of the aspect ratio. There is no clear relationship between varying aspect ratio and the flying qualities of the aircraft. Aspect ratio's 2,3, and 5 seem to form an increasing path, yet aspect ratio of 4 is clearly an anomaly, as is aspect ratio 7. This type of analysis would be useful when aspect ratio is used as a constraint on the preliminary design. It would only be necessary to ensure that the aspect ratio given provides a Level 1 aircraft.

In both cases, the analysis was extremely sensitive to center of gravity location, more so than with the stability derivatives. Because of this sensitivity, this tool is recommended for use in identifying trends, rather than to force the optimization to a specific CAP value.

CHAPTER 6

Conclusions and Recommendations

A comprehensive workstation-based tool to facilitate the optimization of aircraft for handling qualities was designed and implemented. PREDAVOR rapidly calculates stability and dimensional derivatives given a three dimensional model of an aircraft. It then estimates the handling qualities metric control anticipation parameter (CAP) and plots it via a graphical interface on a CAP plot. In this way it allows the user to rapidly assess aircraft geometry changes and identify trends as they pertain to handling qualities. The aircraft may then be optimized for these qualities.

In general, both the longitudinal and lateral derivatives were predicted well.

The inherent vorlax lattice method has been shown to be extremely sensitive to center of gravity location, as is the CAP calculation. This sensitivity must be noted by the user in order to use the tool effectively. The stability derivatives predicted are well within tolerable ranges for such estimations.

Further research will include the possible implementation of this scheme into an existing optimization and aircraft design package, such as NASA's ACSYNT, in order to allow multidisciplinary optimization, including handling qualities, of aircraft during the preliminary design stage.

Notes

Chapter 1

[1] R. Simms, "CE: Engineering a Change in the Design Process," Aerospace America, April 1993.

[2] A. Mykelbust and P. Gelhausen, "Putting the ACSYNT on Aircraft Design," Aerospace America, Sept 1994.

[3] McDonnell Douglas Company, The USAF Stability and Control Digital DATCOM, Vol 2, "Implementation of DATCOM Methods", AFFDL-TR-79-3032 (St. Louis Missouri: April 1979)

[4] Dept. of Defense, Military Standards, Flying Qualities of Piloted Aircraft, MIL-STD-1797A, 30 Jan., 1990.

Chapter 2

[1] A. Myklebust and P. Gelhausen.

[2] NASA Ames Research Center and Virginia Polytechnic State University, ACSYNT V2.0 Overview and Installation Manual, Jan. 1993.

[3] ACSYNT Manual

[4] ACSYNT Manual

[5] L. R. Miranda, R.D. Elliott, and W. M. Baker, "NASA CR-2865 A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications," NAS11-12972, Dec. 1977.

[6] J. D. Anderson, Fundamentals of Aerodynamics, (New York: McGraw-Hill, 1991)

[7] J. D. Anderson

[8] L. R. Miranda et al.

[9] VORVIEW, computer software, J. R. Gloudemans and Dave Kinney, NASA Ames Research Center (Moffett Field, CA.)

Chapter 3

[1] ACSYNT Manual

[2] Duane McRuer, Irving Ashkenas, and Dunstan Graham, Aircraft Dynamics and Automatic Control (Princeton, New Jersey: Princeton University Press, 1973)

[3] Duane McCruer

[4] J. D. Anderson

[5] Dept. of Defense

Notes, continued

Chapter 4

[1] Todd Whelan, "Development and Implementation of Software for Visualizing and Editing Simulation Input Data", (Master's Thesis, California Polytechnic State University, San Luis Obispo, 1996).

[2] RADIAN Flight Simulator, computer software, Fritz Anderson and Todd Whelan (work in progress, California Polytechnic State University, 1996)

Chapter 5

[1] J. Roskam, Airplane Flight Dynamics and Automatic Flight Controls, PartI (University of Kansas: Roskam Aviation and Engineering Corporation, 1979)

[2] R. K. Heffley and W. F. Jewell, "NASA CR 2144 Aircraft Handling Qualities," Dec. 1972

[3] R. K. Heffley